



The UTrack framework for segmenting and measuring dermatological ulcers through telemedicine



Mirela T. Cazzolato ^{a,*}, Jonathan S. Ramos ^a, Lucas S. Rodrigues ^a, Lucas C. Scabora ^a, Daniel Y. T. Chino ^b, Ana E.S. Jorge ^c, Paulo Mazzoncini de Azevedo-Marques ^d, Caetano Traina Jr. ^a, Agma J.M. Traina ^a

^a Institute of Mathematics and Computer Science, University of São Paulo (USP), São Carlos, Brazil

^b InterlockLedger, São Paulo, Brazil

^c Department of Physical Therapy, Federal University of São Carlos (UFSCar), São Carlos, Brazil

^d Ribeirão Preto Medical School, University of São Paulo (USP), Ribeirão Preto, Brazil

ARTICLE INFO

Keywords:

Dermatological ulcers
Wound measurement
Mobile
Segmentation
Image processing
Telemedicine
mHealth

ABSTRACT

Chronic dermatological ulcers cause great discomfort to patients, and while monitoring the size of wounds over time provides significant clues about the healing evolution and the clinical condition of patients, the lack of practical applications in existing studies impairs users' access to appropriate treatment and diagnosis methods. We propose the *UTrack* framework to help with the acquisition of photos, the segmentation and measurement of wounds, the storage of photos and symptoms, and the visualization of the evolution of ulcer healing. *UTrack-App* is a mobile app for the framework, which processes images taken by standard mobile device cameras without specialized equipment and stores all data locally. The user manually delineates the regions of the wound and the measurement object, and the tool uses the proposed *UTrack-Seg* segmentation method to segment them. *UTrack-App* also allows users to manually input a unit of measurement (centimeter or inch) in the image to improve the wound area estimation. Experiments show that *UTrack-Seg* outperforms its state-of-the-art competitors in ulcer segmentation tasks, improving F-Measure by up to 82.5% when compared to superpixel-based approaches and up to 19% when compared to Deep Learning ones. The method is unsupervised, and it semi-automatically segments real-world images with 0.9 of F-Measure, on average. The automatic measurement outperformed the manual process in three out of five different rulers. *UTrack-App* takes at most 30 s to perform all evaluation steps over high-resolution images, thus being well-suited to analyze ulcers using standard mobile devices.

1. Introduction

Motivation. Several health conditions and diseases demand continuous observation and frequent follow-ups regarding the patients' treatment. Thus, having mechanisms to allow such observation without imposing more stress on patients has been sought by medical specialists. Moreover, appropriate tools will make it possible to service patients with diverse health conditions which may make it difficult for patients to leave home and go to medical facilities. Bedridden patients with low mobility or aggravated comorbidities often present fragile health, thus they should preferably receive home treatment. They are also susceptible to infections, for instance, Sars-CoV-2 (COVID-19) and Influenza A (H1N1). Further, patients from remote communities should be able to access effective care and accurate information about their condition

[18].

Among the conditions that require home care, we focus on chronic dermatological ulcers. Burns, infections, and poor blood circulation are examples of skin ulcer causes [22,33]. Dermatological ulcers can also occur due to various skin conditions and the presence of preexisting diseases, such as diabetes [18,27]. Delays in health appointments for patients with skin ulcers may even result in limb amputation [37]. The wound healing process can be complex, dynamic, and lengthy [13]. Health professionals often debride the wound, inspect the ulcer's healing progress and, when necessary, recommend visits to specialists [37]. Frequent follow-ups of patients with ulcers are essential in order to identify the stages of wound healing and to adapt or change aspects of treatment when needed.

Context. Wound size evolution is an effective healing predictor [7,

* Corresponding author.

E-mail addresses: mirelac@usp.br (M.T. Cazzolato), agma@icmc.usp.br (A.J.M. Traina).

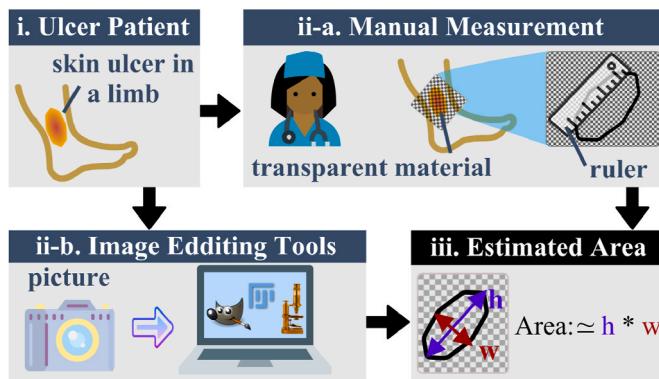


Fig. 1. Manual measurement of dermatological ulcers.

37,39]. Fig. 1 illustrates two traditional and basic procedures for measuring (i) a patient's wound [14,19,32].

The specialist can (ii-a) place a transparent mesh over the patient's limb and mark the wound edge. The mesh usually has squares of one cm^2 , and the wound size is approximated by the number of squares the ulcer covers. The specialist can alternatively (ii-b) take a photo of the wound and a reference gauge (such as a tape, ruler, and adhesive) using a common camera. Then, the specialist uses a semi-automatic image editing tool on a local computer to trace the ulcer edge. In both methods, (iii) the wound area A is given as $A \approx h \times w$, where h and w are the largest sides of the ulcer. However, this methodology is imprecise and can overestimate the ulcer size by more than 70% [2,32].

Difficulties. Works from the literature have proposed computer systems that support the treatment of chronic skin ulcers [2,19,20,25,34]. Some work has investigated the subject from the visual aspects of the image content and the classification of different wound tissues (e.g. necrosis, calluses, granulation, and fibrin) [4,11]. Other studies have focused on the limb segmentation of patients with ulcers [5,34] and also the wound region [11]. Nevertheless, without a user-friendly application, computer-aided solutions are not sufficiently developed to effectively aid the medical practice. Thus, potentially valuable assistance to both the health personnel and patients is lost. Wound care applications must be quick, affordable, accurate, user-friendly, and unobtrusive to the patient to be suitable for everyday use [25,27,37].

Significant effort has been made to accurately analyze photos depicting ulcers using mobile devices [15,27,38]. Mobile Health (mHealth) and telemedicine are effective and essential technologies to provide continuous healthcare support for geographically distant patients. This allows specialists to better monitor patients' conditions and identify the need for suitable intervention. The existing mHealth and telemedicine applications implement supervised classifiers to detect regions with wounds and different tissues in the ulcer. This approach requires a large number of images to both train the model and to allow accurate classification. Unfortunately, there is a lack of large datasets of wound images for model training, which is a significant drawback, leading to models that hardly generalize the classification results [5,34].

Further difficulties with existing applications embodying wound measurement rely on specialized equipment, such as heat sensors and 3D cameras [25]. There are many proprietary tools for wound care, for instance Swift Skin and Wound [2], PointClickCare, +WoundDesk, MOWA.¹ The imitoMeasure² app is free for use but relies on a trained user to inform the correct wound edge, which can overestimate the wound size. The app also has limited resources in its free version, and

the user cannot store ulcer analysis and adequately evaluate the healing evolution. Other tools focus on specific conditions and parts of the body for ulcer prevention, such as heat-based foot ulcer follow-up [27].

Problem Definition and Contributions. This work addresses how to obtain practical and useful measurement as well as follow-up tools for chronic dermatological ulcers. In a preliminary, short work [8] we proposed the *URule-App*, which aids dermatological ulcer measurement employing smartphones. In this work, we significantly extend the initial prototype and introduce the *UTrack* framework.

UTrack allows the user to measure, store, visualize, and share the analysis of chronic dermatological ulcers along a period of time. *UTrack* is a free and easy-to-use solution that can run on a regular mobile device containing a standard camera and without an internet connection. No sensors, special cameras, or other specific devices are required. Therefore, the application is widely accessible to patients, caregivers, and healthcare professionals. The *UTrack* framework includes:

- *UTrack-App*, an Android app for users to take pictures, outline relevant regions, and visualize analysis results in a fast, portable and straightforward manner
- *UTrack-Seg*, an unsupervised method to segment the wound and the measurement object in the images, providing their exterior outline
- A tick marker detection method, which combined with image segmentation, can approximate the wound area in cm^2 or in^2
- A local database to store all analysis for a patient inside the application
- A visualization module to allow users to follow and assess the wound healing evolution
- A report generated by the mobile application for the patient to share with their physician

Recently, Augmented Reality (AR) technology has been employed to measure objects by moving the camera of a mobile device. However, in this study, we employ a ruler-based measuring method because this is the standard protocol followed by health professionals. The use of a ruler or a tape brings analysis confidence to health professionals and can also be double-checked and adjusted whenever necessary. Plus, AR technology can be software and hardware-dependent while the use of a measuring object is accessible and simple (e.g. any reference for one cm or in can be used).

The in-depth experimental analysis shows that *UTrack-Seg* can segment both the ulcer and the measurement reference with high quality. Our approach outperformed the color-based pixel classifiers in average F-Measure by up to 114%, the superpixel-based classifiers by up to 82.5%, the deep learning-based approaches by up to 19%, and the region-growing-based competitor by up to 21%. *UTrack-App* can execute all tasks promptly (up to 30 s for all tasks and high-resolution images) and store all results in a local database. Moreover, *UTrack* can be modified to work with other types of images and wounds. *UTrack-App* is available for download³ under the Creative Commons license.

Paper outline. This paper is organized as follows. Section 2 gives the background. Section 3 describes related work. Section 4 presents the *UTrack* framework. Section 5 details the material and methods employed. Section 6 describes the experimental evaluation. Finally, Section 7 concludes the work.

2. Segmentation of dermatological ulcers

This section covers the relevant background on segmentation methods useful for skin ulcer images. Image I consists of an r by c matrix of $n = r \times c$ pixels $p_i \in I$, such that $1 \leq i \leq n$. A pixel p_i of a colored image

¹ Swift Skin and Wound (<https://swiftmedical.com/>), PointClickCare (<https://pointclickcare.com/>), +WoundDesk (<https://wounddesk.com/>), MOWA (<https://healthpath.com/>).

² imitoMeasure (<https://imito.io/en/imitomeasure>).

³ *UTrack-App* is openly available for download at <http://github.com/mtcazzolato/utrack-app>. The supplementary material provides a demonstration video showing how *UTrack-App* works.

Table 1
Segmentation approaches for dermatological ulcers.

| Work | Approaches | Objective |
|--------------------------|--|--|
| CLMeasure [4] | Classification, Superpixels | Content-based image retrieval |
| ICARUS [11] | Pixel classification, Superpixels and Bag of Visual Signatures | Content-based image retrieval |
| 2PLA [34] | Pixel and Superpixel classification | Limb segmentation |
| QTDU [5] | Superpixels' Deep Learning-based classification | Wound segmentation and measurement (in pixels) |
| DeepLabV3+ [9] | Deep Learning-based segmentation | General purposes |
| ASURA [10] | Deep Learning-based Segmentation | Wound segmentation and measurement (in cm ² or in ²) |
| GrowCut and GrabCut [25] | Region-growing segmentation | 3D wound segmentation and measurement (in cm ² or in ²) |

represented in the RGB color space is a triple $p_i = (R_i, G_i, B_i)$, where R_i , G_i and B_i represent the intensity of p_i respectively for the red, green and blue color channels. A segmentation algorithm sorts every pixel into disjoint subsets [17], which identifies the relevant and non-relevant image regions. For instance, dermatological pictures can be divided into wound and not-wound regions of pixels. Other approaches can also segment wound regions according to specific healing tissues, such as necrosis, fibrin, and granulation. Segmented regions usually present uniform visual characteristics. The algorithms can rely on discontinuity, abrupt changes, and intensity to identify the regions' edges, segmenting the image into groups of similar pixels, following a given criterion [1, 17].

Segmentation algorithms rely on different heuristics to classify images. Table 1 summarizes recent approaches from the literature employed for ulcer images' segmentation. We explain the studies as follows.

2.1. Pixel and superpixel-based segmentation

Superpixel-based approaches group pixels into perceptually meaningful atomic regions [1]. Algorithms generally use color information to group visually similar pixels. The algorithm divides the image I into s disjoint regions, called superpixels. One of the main advantages is that superpixels can replace the rigid structure of the pixel grid of an image, and the generated regions adhere to visual boundaries.

Many studies have explored the superpixels approach to segment skin ulcer images. In Ref. [4] the authors proposed CLMeasure, which employs superpixels to divide and classify the ulcer tissues in the images. Having the superpixels, the features from each region are extracted and labeled using off-the-shelf classifiers such as SVM, Naive Bayes, and Random Forests. ICARUS was proposed in Ref. [11]. It combines superpixels and bags of visual signatures to identify the different patterns depicted in skin healing stages. Both CLMeasure and ICARUS use the extracted features to perform content-based image retrieval. The

2PLA (Two-Phase Learning Approach) method [34] performs pixel and superpixel-based classification to segment a wound. To estimate the wound size, 2PLA requires the user to provide the number of pixels inside the segmented wound, which is the proportion of wound pixels against all pixels in the limb.

2.2. Deep learning-based segmentation

Recently, a significant effort has been made to take advantage of the deep learning algorithms in image segmentation and classification. Deep Convolutional Neural Networks (CNN) has been used mainly in image recognition tasks [35] and can perform semantically rich segmentation. CNN uses convolution operations to encode the input image into a feature map, where the objects are segmented. Blanco et al. [5] proposed QTDU, combining deep learning approaches to classify the superpixels previously extracted from the images according to different ulcer tissues (e.g. granulation and necrosis). QTDU uses ResNet and InceptionV3 CNNs to segment and classify superpixel regions. It receives the superpixels as input and processes six additional layers after using CNN in order to assign the superpixel label. QTDU also estimates the wound area considering the number of pixels inside the segmented wound but not in real-world units.

DeepLab [9] is another deep learning model for image segmentation. It employs *atrous* convolutions to upscale the low-level features and enlarges the field of view of filters with a simple architecture. DeepLabV3+ [9] is its latest version, which has an effective decoder to refine the segmentation, replacing the max pooling operations with depth-wise separable convolutions. The DeepLabV3+ decoder concatenates the encoded features, which are up-scaled by a factor of four, with the corresponding features.

U-Net [30] is an Encoder/Decoder FCN that overcomes the limitation of depending on large amounts of data to train and instead deals with smaller training sets. U-Net improves the segmentation results by considering the spatial information lost in the encoder, copying the activation layers from the decoder, and concatenating them with the up-scaling tensor. In Ref. [10] the authors employ U-NET to propose ASURA, a segmentation approach for ulcer segmentation.

Deep learning approaches have shown highly accurate segmentation results. However, such approaches are supervised and usually require extensive training and data augmentation operations to achieve high classification quality.

2.3. Region growing segmentation

Methods for semi-automatic object segmentation have shown promising improvements in recent work. This is the case of the so-called **region-growing** approaches, which are unsupervised semi-automatic segmentation algorithms. Methods from this category usually rely on manual outline over the input image to perform segmentation. For example, Pavlović et al. [25] proposed a semi-automatic segmentation method based on GrowCut [36] and GrabCut [31] to segment skin ulcer

Table 2
Overview of existing approaches for dermatological ulcer analysis and the proposed method.

| Work | Year | Wound size | No special equipment | Mobile | Available app. | Patient history | Temporal analysis | Info. Sharing |
|-------------------------|------|------------|----------------------|--------|----------------|-----------------|-------------------|---------------|
| Wang et al. [38] | 2015 | × | × | ✓ | ✗ | ✓ | ✓ | ✗ |
| CLMeasure [4] | 2016 | × | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Pasero et al. [23,24] | 2017 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Fraiwan et al. [15] | 2018 | × | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ |
| ICARUS [11] | 2018 | × | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| MyFootCare [6,27] | 2018 | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| Shirley et al. [33] | 2019 | × | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Dastjerdi et al. [22] | 2019 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| QTDU [5] | 2020 | × | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| ASURA [10] | 2020 | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ |
| Proposal: UTrack | 2021 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

images. The method calculates both 2D (i.e., perimeter and area) and 3D (i.e., volume) measurements of a dermatological ulcer, based on interior and exterior manual outlines. CleverSeg [28], which requires only the exterior outline of the object being segmented, uses semi-automatic segmentation over grayscale images. CleverSeg, initially developed for segmenting paraspinal muscles in magnetic resonance images, uses mathematical morphology [17] to estimate the target region. It receives the image and the external outline of the object of interest in the image. It is shown [28] that CleverSeg outperforms existing region-growing segmentation approaches, such as GrowCut and GrabCut.

In this work, we describe our novel *UTrack-Seg* approach, an improved version of the CleverSeg [28] method, which was developed to process grayscale images. *UTrack-Seg* extends CleverSeg to handle the RGB color channels required to process the colored photos of dermatological wounds and segments the wound and ruler regions in the images, while considering the provided exterior outline. Importantly, *UTrack-Seg* is unsupervised and requires no training step to segment images.

3. Related work

This section presents related work regarding computational methods to analyze dermatological ulcer images in practical approaches. We cover recent methods involving the aspects related to our proposal, summarized in Table 2, as follows:

- *Wound size*: whether the approach measures the ulcer size in the image
- *No special equipment*: whether collection and analysis of images is independent of special equipment
- *Mobile*: whether it is available as a mobile solution to be used in practice
- *Available application*: whether it is free for patients and health practitioners
- *Patient history*: whether it keeps historical information and analyses
- *Temporal analysis*: whether it tracks the evolution of a patient's condition over time
- *Information sharing*: whether information can be shared among patients and health professionals

Wang et al. [38] proposed a wound image analysis system for Android. The system provides real-time wound segmentation, which employs a mean shift algorithm to determine the wound boundaries. The system calculates the wound area with a simple connected-region detection method and predicts the wound healing status based on a red-yellow-black color evaluation model. The system requires a particular device named Image Capture Box to acquire images from patients. The device has mirrors and white LEDs to provide consistent camera angle and lighting. The generated mask determines the wound's boundaries between regions of pixels classified as granulation, slough, or necrosis. Storing data along time allows the system to identify trend patterns and monitor the patient's healing evolution. However, the system has some disadvantages. It gives the wound size as the sum of pixels and not an actual measurement unit like cm^2 or in^2 . Also, it does not evaluate or report the quality of segmentation and classification.

CLMeasure [4] focuses on retrieving skin ulcer images based on their visual similarity, exploiting the different healing stages which are depicted in the image. The method subdivides the image into superpixel regions and classifies each superpixel considering the features that indicate the classifications of granulation, necrosis, callous, and fibrin. It uses a composition function over the labeled regions for distance assessment among images. ICARUS [11] is based on superpixels

combined with Bags of Visual Signatures. It focuses on content-based retrieval of ulcer images and presents higher quality results than CLMeasure [11] when considering skin ulcers at different healing stages.

Pasero et al. [23,24] proposed an automatic method for leg ulcer segmentation and analysis. Photos are acquired using a smartphone camera, and the segmentation is based on a morphological opening with a disk-shaped structuring element, over both RGB and HSV color spaces. The authors estimate the wound size using a specific ruler to convert the number of pixels to cm^2 . The automatic measurement of wounds presented results comparable to manual measurements. However, the study does not present further analysis nor results of comparisons with other segmentation approaches.

Fraiwan et al. [15] proposed a mobile app for foot skin ulcer analysis using a thermal camera for temperature-based image processing. The *mULCER* mobile app [26] analyzes images and additional information and provides informative texts to the user. Marchione et al. [21] explains that existing approaches regarding software for pressure ulcer prevention most commonly rely on specialized equipment and sensors, strict control of the patient's position in bed, and written information.

MyFootCare [6,27] is a mobile application prototype that provides self-care support to people with diabetic foot ulcers. The app focuses on engaging patients in self-care practices by allowing them to track the wound's healing process based on photos taken by a mobile phone. It runs on Android devices using a Java Framework and uses the OpenCV library for wound segmentation and analysis. The segmentation requires the user to roughly draw a circle around the ulcer to denote the healthy skin tissue edges and another inside the ulcer. The app runs OpenCV's Morphological Watershed algorithm to segment the image background and isolates the ulcer from healthy tissues. It relies on a small green sticker of 1 cm diameter placed next to the wound to scale and convert the wound area to cm^2 units. MyFootCare records the wound area and information about how the patient feels and what they are doing and allows following the wound size evolution. MyFootCare is not publicly available, and the work does not present experimental evaluations about segmentation nor wound area calculation. As it relies on a specific sticker, it is not useful for untrained patients.

Shirley et al. [33] proposed a process to measure the length, area, and volume of wound ulcers. It processes a sequence of images captured by a smartphone and creates a 3D representation using the Structure-from-Motion (SfM) algorithm. It calculates the wound dimensions based on a dynamic threshold in the CIELAB color space. Although acquiring the photos through a smartphone, the subsequent segmentation and calculations are not performed on the device. Moreover, it does not consider temporal data from the analysis, and the calculated wound dimensions are not in real-world units.

Dastjerdi et al. [22] proposed another method for semi-automatic wound segmentation and area measurement. It utilizes both 2D and 3D representations, processing a single photo or a video, respectively. The 2D photo can be taken with a digital camera or smartphone, with a flexible paper ruler placed near the wound for size measurement. The segmentation starts by roughly outlining the region of interest around the ulcer. Then, a trained Random-Forest (RF) calculates a probability map of each pixel belonging to the wound or healthy skin. By employing the Otsu's threshold, a binary mask containing the wound area is created over the probability map. The ruler is segmented to calculate the ratio between pixels and centimeters. However, the method was not compared to other segmentation tasks nor wound area calculations, and it does not track the wound size evolution.

QTDU [5] is another deep learning-based method that segments the ulcer regions through superpixel-based classification. Besides being computationally costly, the wound area is the sum of pixels in the

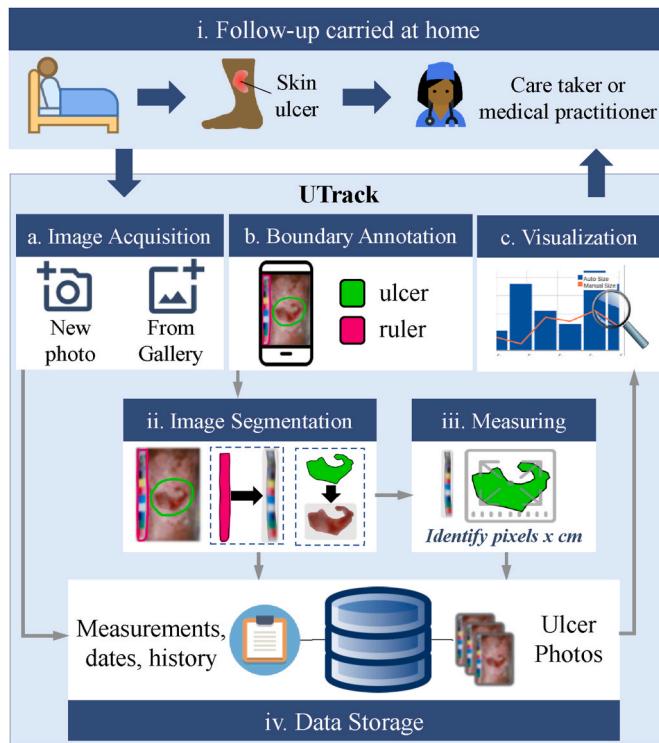


Fig. 2. The proposed framework *UTrack*.

segmented region. Finally, the ASURA system [10] was proposed for segmentation and ulcer area measurement in real-world units (cm^2). A U-NET network performs the segmentation, which has shown to be highly accurate. ASURA automatically measures the size of the ulcer, and it is complemented by a desktop application to adjust the measurement manually. However, the user has to take the photo and transfer it to a local computer to segment and estimate the ulcer size. This pipeline may impair the use of ASURA in the daily medical procedures

carried out at patients' homes.

Most approaches described in this section require extensive training to classify and segment the wound regions. However, it is difficult to obtain large collections of dermatological ulcer images for training. Moreover, supervised classifiers and content-based retrieval approaches need to consider privacy protection of the patients' images used to train models and to show similar images, respectively. In the following, we detail our proposed *UTrack* framework to provide adequate support for caregivers and health professionals in their daily routine.

4. The proposed framework

In this work, we present our *UTrack* framework for supporting the follow-up of chronic dermatological ulcers. *UTrack* provides modules to segment the ulcer and the measurement object, estimate the ulcer area, keep historical data, and visualize the wound and symptoms evolution. Fig. 2 presents the main modules of the *UTtrack* framework. We designed *UTrack-App* to make the *UTtrack* features available for patients, caregivers, and healthcare professionals and to support ulcer patients' home care (Step i). *UTtrack-App* assists users (a) to acquire a photo of the ulcer, (b) mark the ruler and wound boundaries, and (c) provide visualization of the analysis results and the historical records. Each outlined region is automatically segmented (Step ii) with our proposed segmentation method *UTtrack-Seg*. The next subsections detail *UTtrack* and its components.

4.1. *UTtrack-App*

UTtrack-App is an Android implementation of *UTtrack*. Fig. 3 shows screenshots taken from the main screens of the app. In the main screen (a) users can analyze a new image, check the patient's history, track symptoms and wound size evolution, and obtain information about the app. The app can keep track of one or more ulcers (b). The user can utilize a picture from the device's gallery or take a new photo (c) and inform related symptoms. The green pen must be used to mark the ulcer outline and the magenta pen for the ruler (d). Outlines must be closed boundaries, and segmentation works well even with coarse contours. The analysis is performed after the segmentation, and *UTtrack-App* shows the ulcer region size in cm^2 or in^2 (e). The user can opt to manually revise

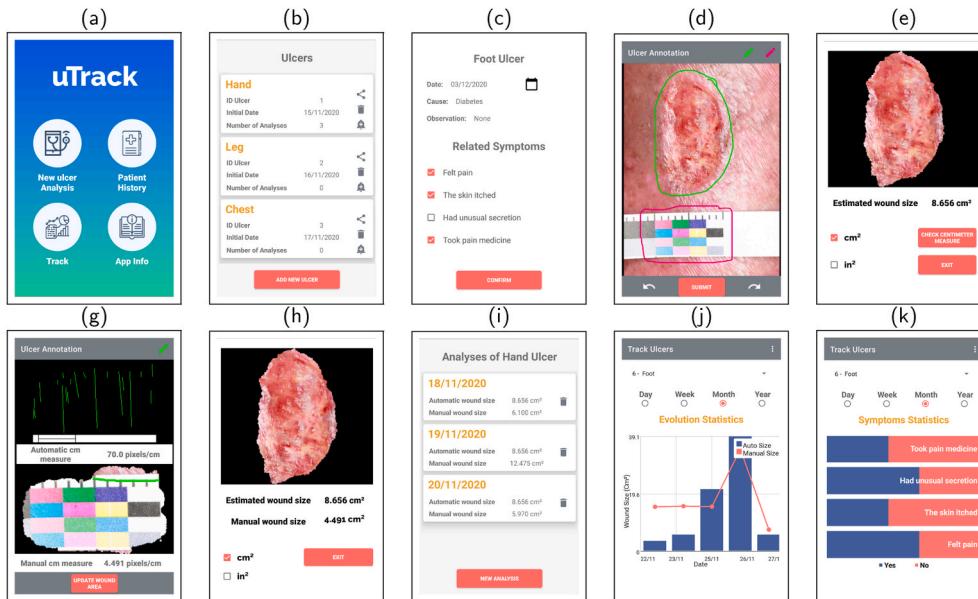


Fig. 3. *UTtrack-App* screenshots.

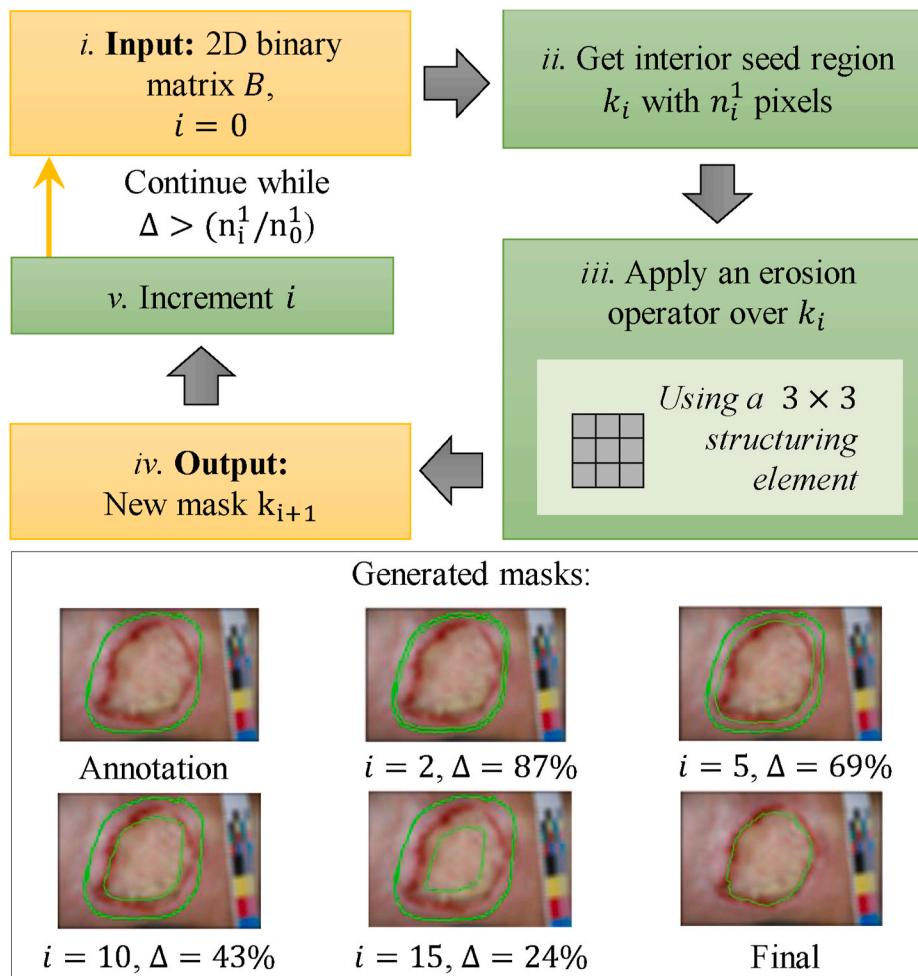


Fig. 4. UTrack-Seg erosion operations to estimate the interior seed region.

Table 3
Specification of smartphones used in the experiments.

| S1. Xiaomi Mi 8 Lite (2018/3) | |
|---------------------------------|---|
| OS | Android 10 MIUI 11 |
| Processor | 4 × 2.2 GHz Kryo 260 + 4 × 1.8 GHz Kryo 260 |
| GPU | Adreno 512 |
| RAM | 4 GB |
| S2. Samsung Galaxy A51 (2019/4) | |
| OS | Android 10 Samsung One UI 2.0 |
| Processor | 4 × 2.3 GHz Cortex-A73 + 4 × 1.7 GHz Cortex-A53 |
| GPU | Mali-G72 MP3 |
| RAM | 4 GB |
| S3. Redmi Note 9 Pro (2020/1) | |
| OS | Android 10 MIUI 11 |
| Processor | 2 × 2.3 GHz Kryo 465 Gold + 6 × 1.8 GHz Kryo 465 Silver |
| GPU | Adreno 618 |
| RAM | 6 GB |

the wound measurement (g and h) by informing the measure of a centimeter in the photo (h). Then, the app recomputes the ulcer size and displays the new result.

For every registered ulcer, (i) UTrack-App keeps the respective analysis in the local database. The user can see the wound healing evolution in terms of wound size and symptoms frequency (j and k). Notice in (b) that the user can share a report of each registered ulcer. The app will generate a *pdf* file with all analysis results and visual evolution of the corresponding ulcer. This report allows patients to share their information with health professionals.

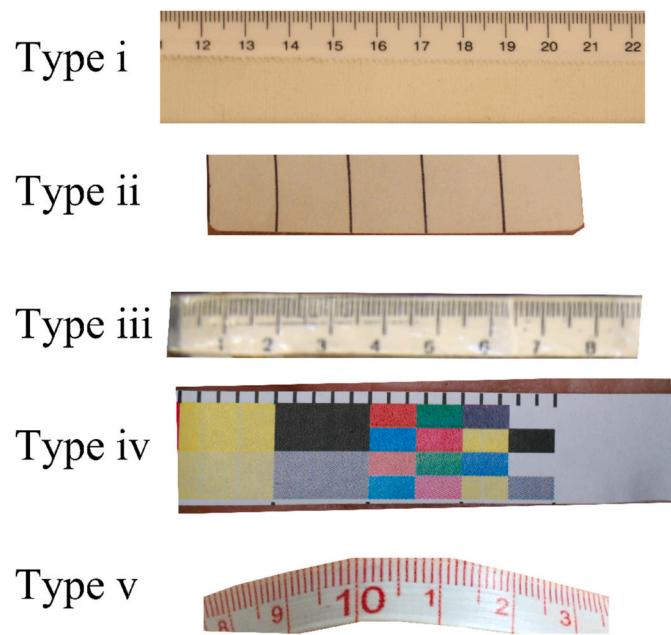


Fig. 5. Five types of measurement tools evaluated.

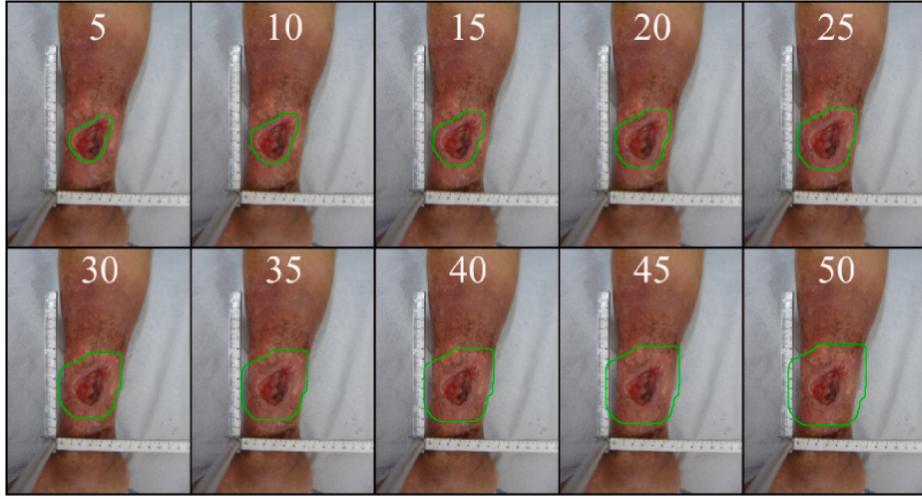


Fig. 6. Examples of external outlines (in green) at different distances (informed by the white numbers) to the wound edge.

4.2. UTrack-Seg: image segmentation

UTrack-Seg considers the foreground segmentation, which can be the ulcer or the measurement tool, as well as the background. Images from mobile devices are taken as color images in the RGB (Red, Green, Blue) color model, represented as a matrix I with r rows and c columns: (there are $n = |I| = r \times c$ RGB pixels p). The outlines are set in another matrix, Seeds L , which has the same dimensions as I . *UTrack-Seg* has three main steps, as presented in [Algorithm 1](#) and detailed as follows:

Algorithm 1. UTrack-Seg

Algorithm 1: UTrack-Seg

```

Input:  $I$ : Image.  $L_{in}$ : exterior annotation/labels matrix.
Output:  $L_{out}$ : Segmented image or labels
// 1. Interior Annotation Estimation
1  $B \leftarrow \text{binarize}(L_{in})$ 
2 while  $k_i^1/k_0^1 > \Delta$  do
3    $B \leftarrow \text{erode}(B)$ 
4  $L_{out} \leftarrow \text{joinLabels}(B, L_{in})$ 
// 2. Weighted Matrix Initialization
5  $m \leftarrow 255$ 
6 if  $l_i \neq 0, \forall l_i \in L_{out}$  then
7    $w_i \leftarrow 1.0$ 
// 3. Segmentation
8 for  $\forall p_i \in I$  and  $\forall p_i^j \in I, j = \{1, 2, 3, \dots, 8\}$  do
9    $h \leftarrow m - \text{diff}(p_i, p_i^j)$ 
10   $s \leftarrow w_i \times h/m$ 
11  if  $(s - w_i^j) > \theta$  then
12     $w_i^j \leftarrow \text{mean}(s, w_i^j)$ 
13     $l_i^j \leftarrow l_i$  from  $L_{out}$ 

```

1. **Interior outline estimation:** After the user outlines the wound and the ruler, both as closed boundaries, *UTrack-Seg* starts estimating the interior of each region. We represent the matrix L as L_{in} (input) and L_{out} (output, the modified matrix) in the algorithm. First, the method fills every pixel inside the outline in the input seeds matrix L_{in} with ones. Next, the algorithm transforms the image (L_{in}) into a new 2D binary matrix B (temporary), where n_1 interior pixels (white/ones) receive values 1, and the remaining n_0 pixels receive values 0, such that $n_0 + n_1 = n$ (line 1). Then, as depicted in [Fig. 4](#), *UTrack-Seg* iterates over B using mathematical morphology.

On each iteration, the image is eroded by a 3×3 ‘square’ structuring element (line 3). The iteration continues until the relation of pixels in the eroded (k_i^1) and initial (k_0^1) masks is lower than Δ , which is computed

from measurements of the image, as will be explained later. The estimated interior outline is taken as the eroded region (line 2). Entries of the outlines’ matrix L_{out} have values equal to 0 (unlabeled), 1 (foreground or interior region) or 2 (background or exterior outline) (line 4).

2. **Weight matrix initialization:** Based on the label matrix L_{out} , the algorithm initializes a weight matrix W to fill every entry $\forall w_i \in W$ with ones for corresponding labeled entries $l_i \in L_{out}$ (interior and exterior outlines) and with zeros otherwise. The maximum pixel intensity m has the value of 255, considering that the RGB values on each channel are in the range $[0, 255] \in \mathbb{Z}$.
3. **Segmentation:** Let us consider each pixel $p_i \in I, i = \{1, \dots, n\}$ and its 8 neighbor pixels $p_i^j \in I, j = \{1, \dots, 8\}$ (line 8). First, *UTrack-Seg* calculates the absolute difference (h) between the RGB pixel intensity (p_i) and its neighbors’ (p_i^j) and subtracts the result by the maximum intensity (m) (line 9). Then, h is normalized by m and multiplied by the strength w_i , resulting in strength s (line 10). Finally, the method computes the difference between s and w_i^n (which is the current strength of the neighbors). If this difference is greater than θ (line 11), *UTrack-Seg* averages the strength w_i^n with s , which is the new strength (line 12), and updates the corresponding label l_i^j (line 13).

In our method, the Euclidean Distance is used to calculate the distance among pixels (line 9). However, as a subject for future evaluation, other distance metrics can be considered as well. *UTrack-Seg* stops repeating Step 3 after a maximum number of iterations or when the algorithm converges. We empirically set θ (line 11) to 1% to avoid averaging values that only change after the second decimal place. We automatically set the Δ threshold used in Step 1 by averaging the Eccentricity and Solidity [29] measures of the wound. The segmented regions are the input of *UTrack*’s next step.

4.3. Tick detection and area measurement

As depicted in [Fig. 2](#) - Step iii, *UTrack* detects the ticks on the segmented ruler or tape obtained by *Utrack-Seg* ([Fig. 2](#)-Step ii) to compute the area of the ulcer. *UTrack* starts cropping the Minimum Bounding Rectangle (MBR) of the region of interest from matrix M using the segmented measurement object. It is assumed that the image is in the horizontal orientation, but it can be rotated if needed. The image is converted to binary and grayscale. Then, *UTrack* acquires the best threshold applying the ISODATA [3] algorithms over the image. The method finds the ruler ticks by combining the Line Segment Detector (LSD) approach [16] with a vertical edge filter. The ruler ticks are

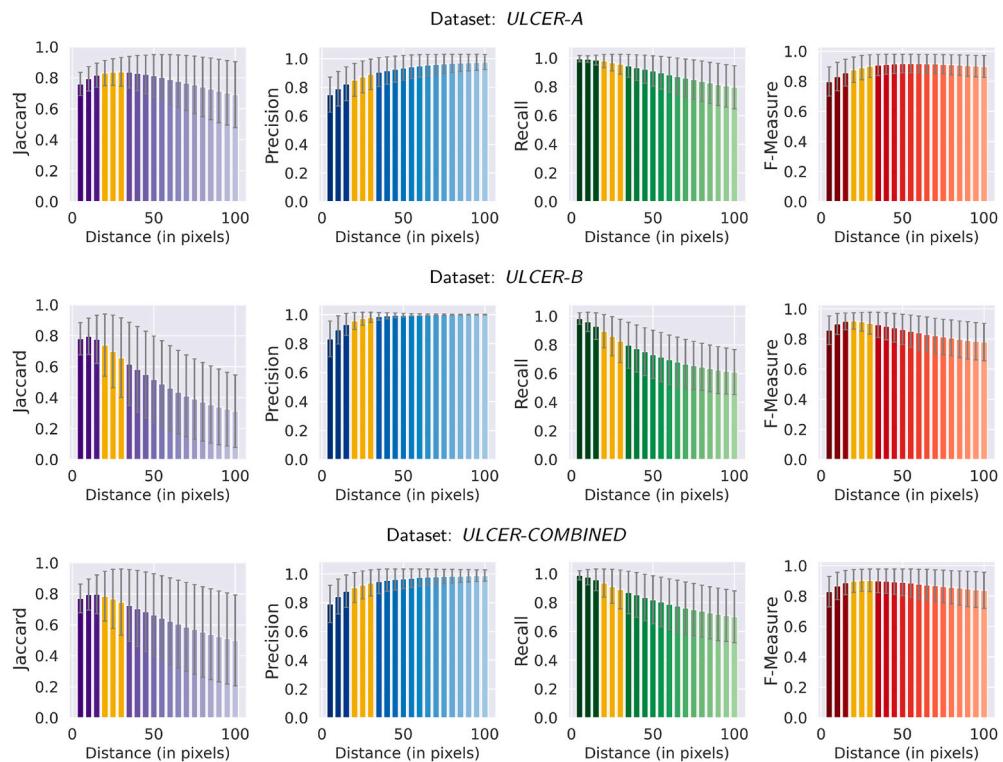


Fig. 7. *UTrack-Seg*: Segmentation quality according to the distance of the outline boundary to the ulcer edge. The yellow bars show the distances presenting the best trade-off between precision and recall.

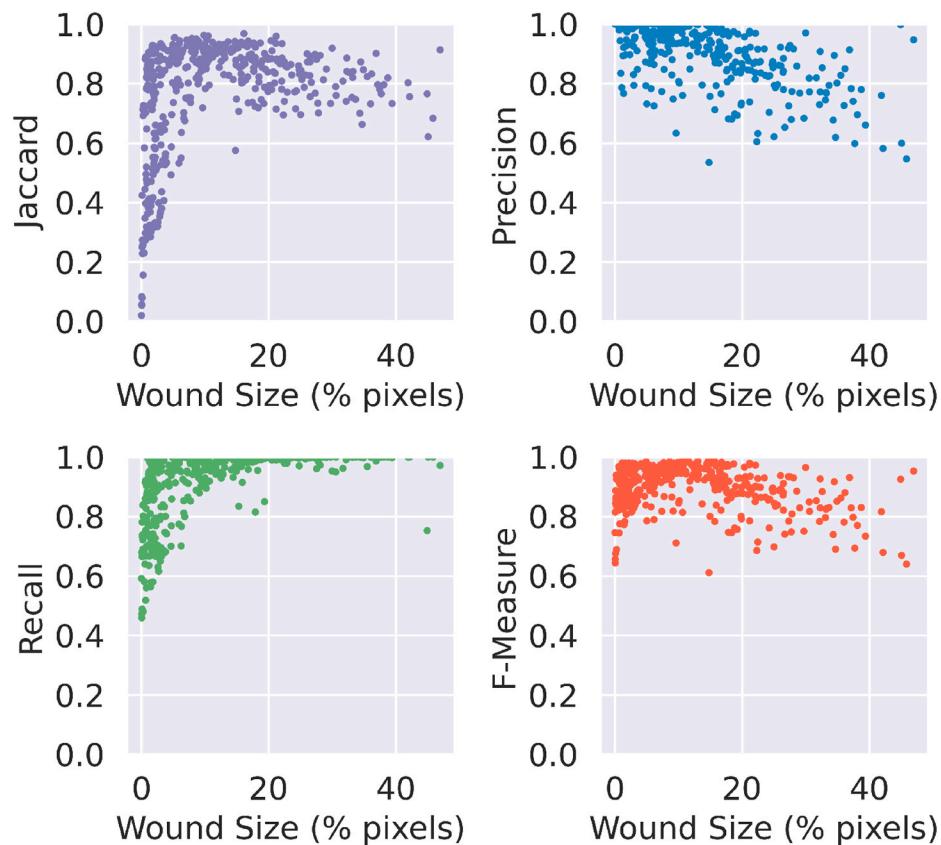
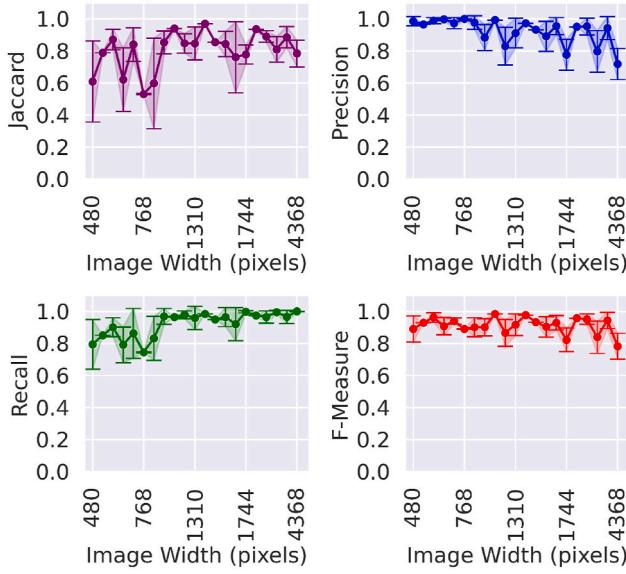


Fig. 8. Segmentation quality according to the wound size, considering the dataset *ULCER-COMBINED*, where each dot corresponds to an image: the higher the better.

(a) Segmentation quality per image resolution, considering the dataset *ULCER-COMBINED*



(b) Segmentation quality per patient and image resolution: every line corresponds to an individual patient from dataset *ULCER-A*

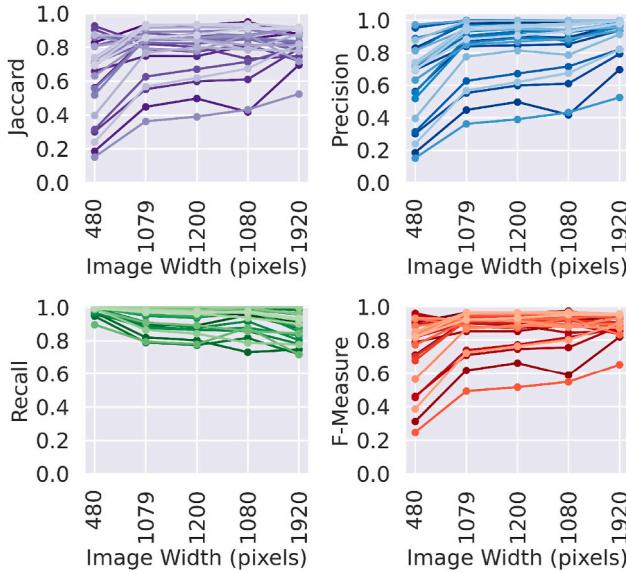


Fig. 9. Segmentation quality per image resolution considering (a) the entire dataset (*ULCER-COMBINED*) and (b) the images from every patient (*ULCER-A*).

grouped according to their angles. The algorithm keeps the segments with an angle difference of at most 5° in relation to the group with more elements, i.e. the most parallel ticks. The remaining ticks are grouped by size, and for each group *UTrack* calculates the distance between the ticks. The distance with the highest probability is given as the proportion of pixels per centimeter.

Manual Correction of a Length Unity. The visual patterns of distinct measurement objects can present patterns (or elements) such as ticks or markers. The ticks' detection can be presented in low quality according to different measurement tools, e.g. transparent rulers or rubber/plastic tapes that can be folded and distorted during the photo acquisition. To tackle the problem of incorrect length estimations, *UTrack* allows the user to insert the measurement of a centimeter by hand (see the

horizontal green line depicted in Fig. 2 (g)). Considering n_w as the number of pixels in the ulcer region (i.e. the segmented wound), and n_r as the pixel density, i.e. the proportion of pixels/cm in the measurement object. The ulcer area A in cm^2 is calculated as $A = n_w/n_r^2$, and the area in in^2 is computed as $A = n_w/(n_r/2.54)^2$. *UTrack* informs the ulcer size in cm^2 or in^2 and shows the segmented region on the screen (Fig. 2-vi).

5. Material and methods

In this section, we detail the setup, datasets, and algorithms employed to evaluate *UTrack*.

5.1. *UTrack-App: implementation and setup*

UTrack-App is an Android application, implemented in Kotlin, and is self-contained and designed to perform all *UTrack*'s mobile operations. We followed the usability guidelines and best design practices provided by Google's Material Design⁴ to develop visual components for Android applications. The app incorporates two external APIs, both implemented in Java and widely used in the development of mobile applications: MP Android Chart⁵ for graphs visualization, and AndroPDF⁶ for report generation. *UTrack-App* was developed using Android Studio IDE. Both *UTrack-Seg* and the tick detector algorithm were developed in Java, using the OpenCV 3.4.5 library. The minimum requirements to install *UTrack-App* are Android OS 8.0 (Oreo - 2017) or higher and at least 1 GB of RAM.

Elapsed times of *UTrack*'s segmentation tasks were collected by simulating the application using Android OS 9.0 (Pie - 2018) in the Android development IDE itself. We also collected the real execution time of *UTrack-Seg* running on three different smartphones, detailed in Table 3. During the experiments, the smartphones were in airplane mode, with app notifications and synchronization disabled to avoid having other processes consuming the devices' processing time.

5.2. Datasets and segmentation approaches

The evaluation step was performed over two datasets, *ULCER-A* [10] and *ULCER-B* [12]. Each dataset has 213 photos of chronic dermatological ulcers taken from patients' lower limbs. The datasets contain arterial and venous ulcers, at different sizes and healing stages (fibrin, granulation, necrotic and callous) as well as different skin colors and treatments. Also, the images have different resolutions. *ULCER-A* has images taken from 23 patients using no protocol to define the background or camera distance to the wound. Each patient has a sequence of photos taken spanning 90 days. *ULCER-B* has images taken following a predefined protocol (using the same camera, blue background, and same measurement tape). The ground truth was obtained from experts who manually segmented the images. They used an image editing tool to precisely contour the wound and the measurement object regions. The contours were used as masks to evaluate both *UTrack-Seg*'s and the GrowCut's segmentation quality. For further information on the datasets, refer to Refs. [10,12].

We compared *UTrack-Seg*'s segmentation results with seven other methods: GrowCut, which is a well-known region-growing approach; ASURA, QTDU, and DeepLabV3+, which are Deep Learning-based wound segmentation methods; CLMeasure and Icarus (superpixel-based), and Color Classification (pixel-based), which extracts color features from images and classifies them using simple classifiers. ASURA's trained models (on *ULCER-A*, *ULCER-B*, and *ULCER-COMBINED*) and implementation were obtained from the original work [10]. We

⁴ <https://material.io>.

⁵ <https://github.com/PhilJay/MPAndroidChart>.

⁶ <https://github.com/ch0nch0l/AndroPDF>.

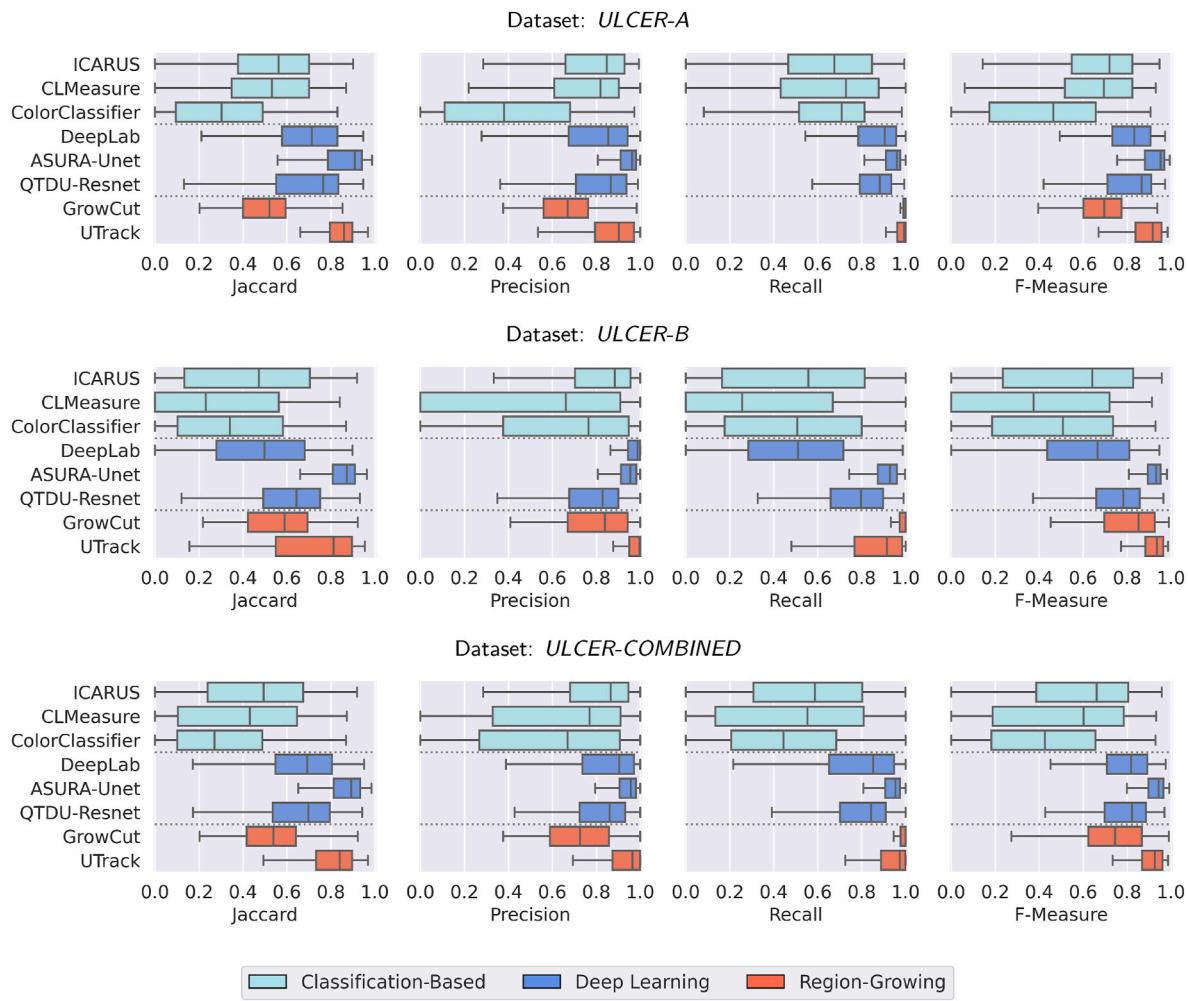


Fig. 10. Segmentation quality of ulcer regions: the higher the better.

Table 4

Segmentation quality results for both datasets (ULCER-COMBINED).

| Approach | Jaccard | std | Accuracy | std | Precision | std | Recall | std | F-Measure | std |
|-----------------|-----------------|-------------|-----------------|-------------|-----------------|-------------|-----------------|-------------|-----------------|-------------|
| ICARUS | 0.459983 | 0.26 | 0.940720 | 0.05 | 0.746801 | 0.30 | 0.542459 | 0.31 | 0.580107 | 0.28 |
| CLMeasure | 0.388531 | 0.28 | 0.932791 | 0.06 | 0.612312 | 0.36 | 0.493973 | 0.35 | 0.496194 | 0.32 |
| ColorClassifier | 0.306654 | 0.23 | 0.887320 | 0.10 | 0.582349 | 0.35 | 0.448197 | 0.29 | 0.423010 | 0.27 |
| DeepLabV3+ | 0.643035 | 0.22 | 0.959489 | 0.04 | 0.832196 | 0.20 | 0.760207 | 0.25 | 0.754917 | 0.21 |
| ASURA | 0.838948 | 0.16 | 0.985020 | 0.02 | 0.915414 | 0.12 | 0.909545 | 0.13 | 0.901250 | 0.13 |
| QTDU | 0.641788 | 0.21 | 0.963014 | 0.03 | 0.789566 | 0.21 | 0.769757 | 0.20 | 0.756853 | 0.20 |
| GrowCut | 0.537223 | 0.16 | 0.767221 | 0.13 | 0.728603 | 0.17 | 0.947223 | 0.15 | 0.743007 | 0.15 |
| UTrack | 0.778915 | 0.17 | 0.897829 | 0.08 | 0.922237 | 0.10 | 0.918869 | 0.11 | 0.905617 | 0.07 |

reimplemented all other algorithms used in our experimental analysis in order to provide a fair comparison among the methods, which share the same platform. The Deep Learning approaches were evaluated using 5-fold cross-validation. We followed the steps described in Ref. [10] to perform Data Augmentation for ASURA's model training and to split the images into the folds in such a way that images from a single patient are only in one fold.

We evaluated the *UTrack*'s tick detection ability using a set of 22 tapes and rulers, classified as types i to v, as depicted in Fig. 5. We compared the accuracy of our application with the manual measurement results obtained by four subjects who are familiar with the issue and used the app to draw a centimeter (cm) in every tool.

6. Experimental analysis and discussion

This section begins the evaluation of *UTrack-Seg*'s segmentation quality according to the distance of the exterior outline to the wound edge with different image resolutions and wound sizes. We compare *UTrack-Seg* results with state-of-the-art segmentation approaches focused on analyzing dermatological ulcers. We show how well *UTrack* performs ticks detection for different measurement tools. Finally, we present a performance analysis of *UTrack-App*, considering the different tasks of the *UTrack* framework.

6.1. *UTrack-Seg* evaluation

Our evaluation begins with the *UTrack-Seg*'s parametrization and the segmentation quality, according to the image and wound sizes.

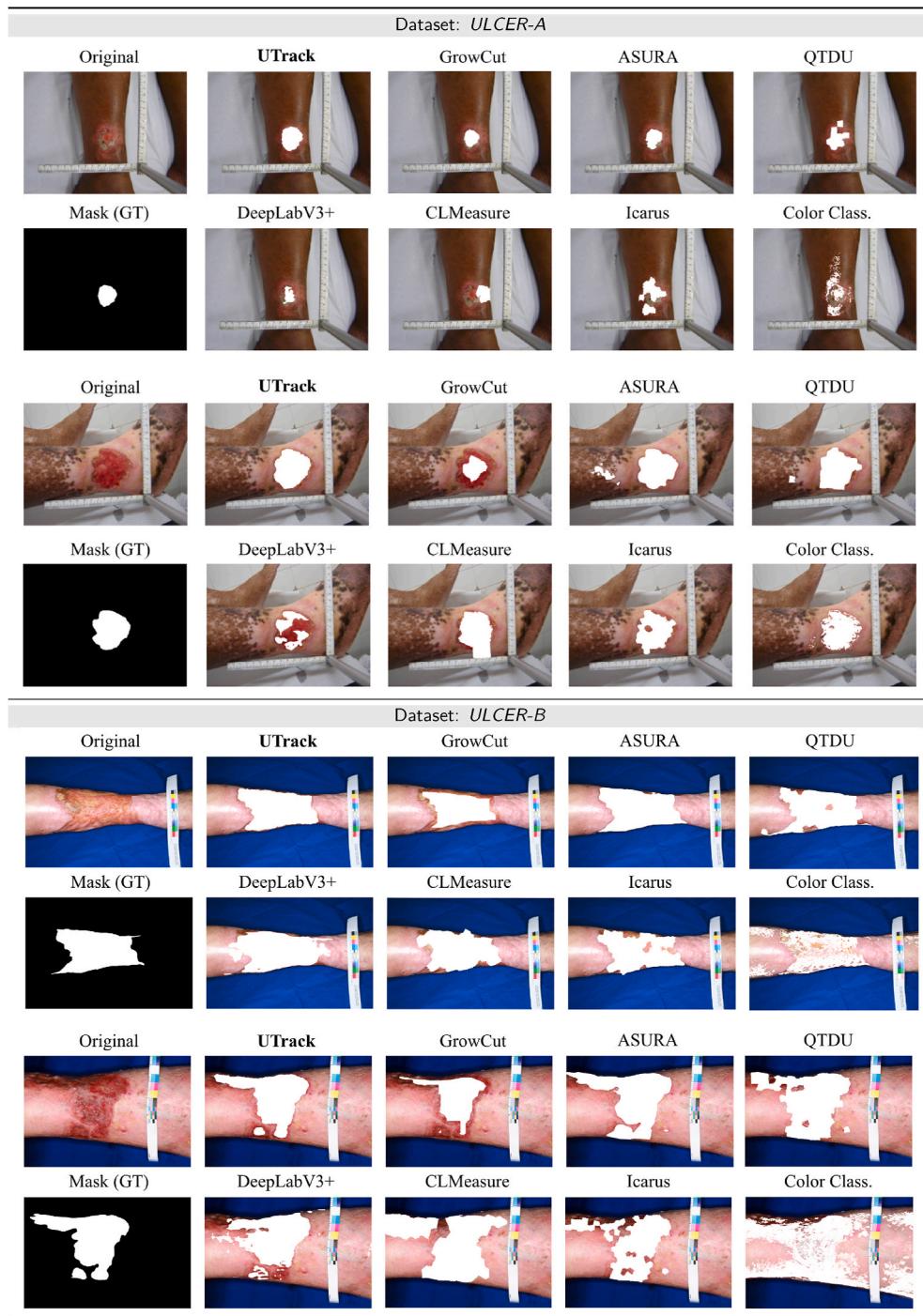


Fig. 11. Segmentation quality: *UTrack-Seg* outperformed its competitors and accurately segmented wound regions.

6.1.1. Distance from outline to wound boundary

This experiment evaluates how the exterior outline distance to the ulcer edge affects the segmentation quality. Fig. 6 shows an example of external outlines at different distances (in pixels) to the wound edge. At distances of up to 15 pixels, the external outline must be carefully provided to better follow the wound edge. Outlines at distances higher than 20 pixels to the wound edge can be provided more freely and less precisely by the users.

Fig. 7 guides the setting of the best distance range for the user to indicate the wound. The charts show the Jaccard, precision, F-Measure (Dice Score), and recall results of *UTrack-Seg* according to the distance (in pixels) of the exterior outline. The horizontal axis shows the distance

range from 5 to 100 in steps of 5 pixels. We evaluate the segmentation of datasets *ULCER-A*, *ULCER-B*, and the combination of both, shown in the graphs as *ULCER-COMBINED*. As a general rule, the farther the outline to the wound edge, the higher the precision and the lower the recall. This occurs because outlines too far from the wound edge tend to consider most of the wound pixels, but they over-segment the wound region, also attaining a high false-positive rate. The best trade-off between precision and recall was obtained with distances between 20 and 30 (highlighted by the yellow bars), as shown by the F-Measure and Jaccard results. Outlines farther than 25 pixels from the wound boundary are finely adjusted by *UTrack-Seg* to obtain accurate results, and we employ this setting in the following experiments.

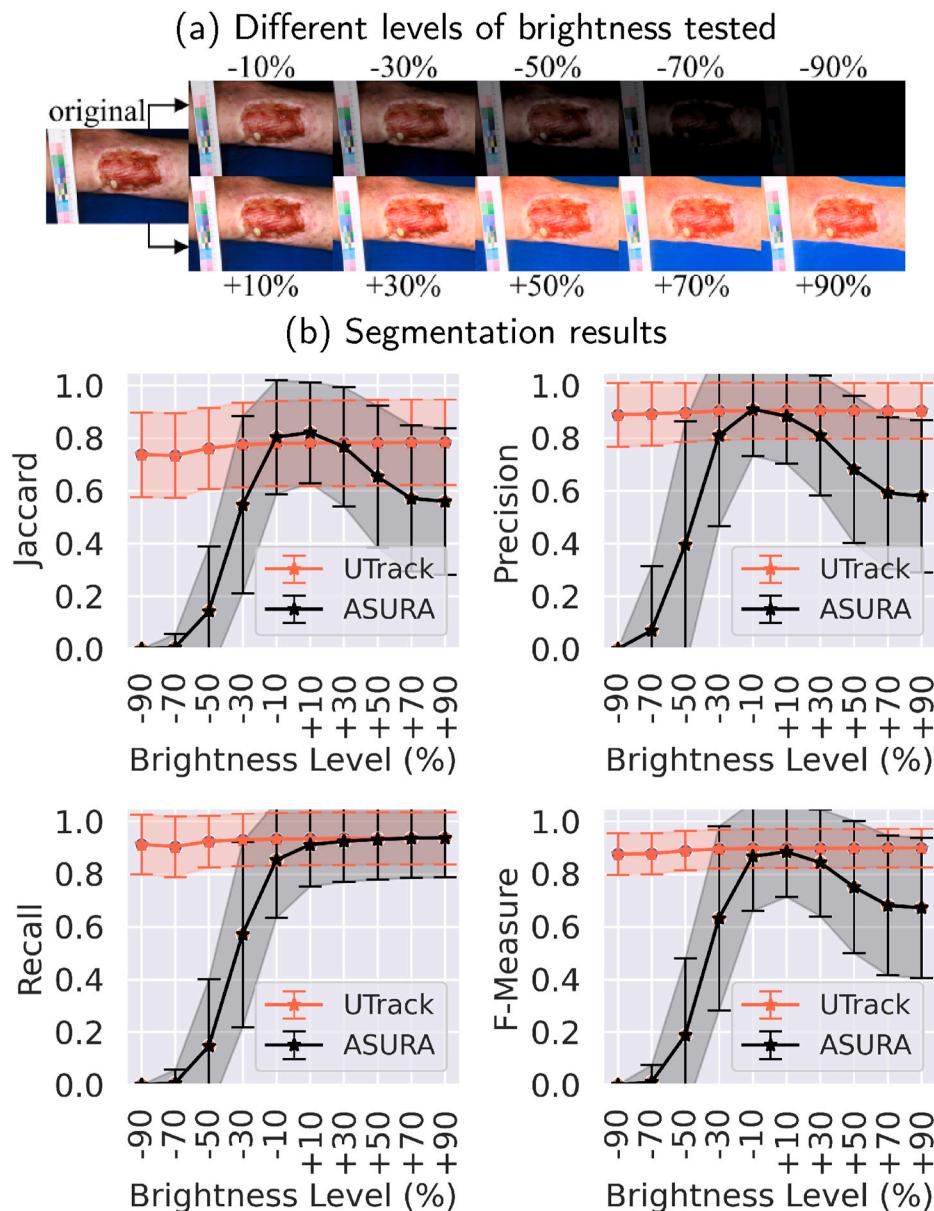


Fig. 12. Segmentation quality according to the image brightness, considering the dataset *ULCER-COMBINED*: *UTrack-Seg* (redish graph) outperforms *ASURA* (blackish graph) at both high and low values of brightness.

6.1.2. Quality per wound and image size

Fig. 8 presents the segmentation quality of *UTrack-Seg* according to the proportion of pixels representing the wound. Tiny wounds (close to zero pixels) incur losses for Jaccard and recall values. However, for the other wound sizes, *UTrack-Seg* produced steadily high segmentation results.

Finally, Fig. 9 (a) presents the segmentation quality of *UTrack-Seg* according to the image resolution existing in both datasets *ULCER-A* and *ULCER-B* (represented here by the image width). Again, *UTrack-Seg* was able to accurately segment the images regardless of the size. This result supports our claim that *UTrack-Seg* successfully works with images acquired by smartphones, even for devices presenting different image resolutions. *ASURA* resizes the input image to a 512×512 matrix. Thus, we did not report results for size variations in this study. Fig. 9 (b) shows the segmentation quality of images from 23 patients (from the *ULCER-A* dataset) considering different resolutions. Each line on the chart corresponds to a patient. Aiming at diminishing cluttering, we employ a color shading variation for the lines in the graph. *UTrack-Seg* obtained fair

results even when segmenting images with the lowest resolution (480×640). Images from five patients presented low-quality results for other resolutions. However, the great majority of lines show high and steady segmentation quality.

6.2. Segmentation quality

Here, we compare *UTrack-Seg* results with state-of-the-art methods focused on segmenting dermatological ulcers. The competitors are from three categories: Classification-based segmentation, Deep learning-based segmentation, and Region-growing segmentation. Fig. 10 presents the Jaccard, precision, recall, and F-Measure (Dice Score) results for *ULCER-A*, *ULCER-B* and their combination *ULCER-COMBINED*. Table 4 summarizes the results for the *ULCER-COMBINED* dataset. Considering the average F-Measure results of the classification-based approaches, *UTrack-Seg* outperformed the color classifier by 114%, IC-ARUS by 56%, and CLMeasure by 83%. *UTrack-Seg* was around 20% better than the Deep Learning methods DeepLabV3+ and QTDU, and

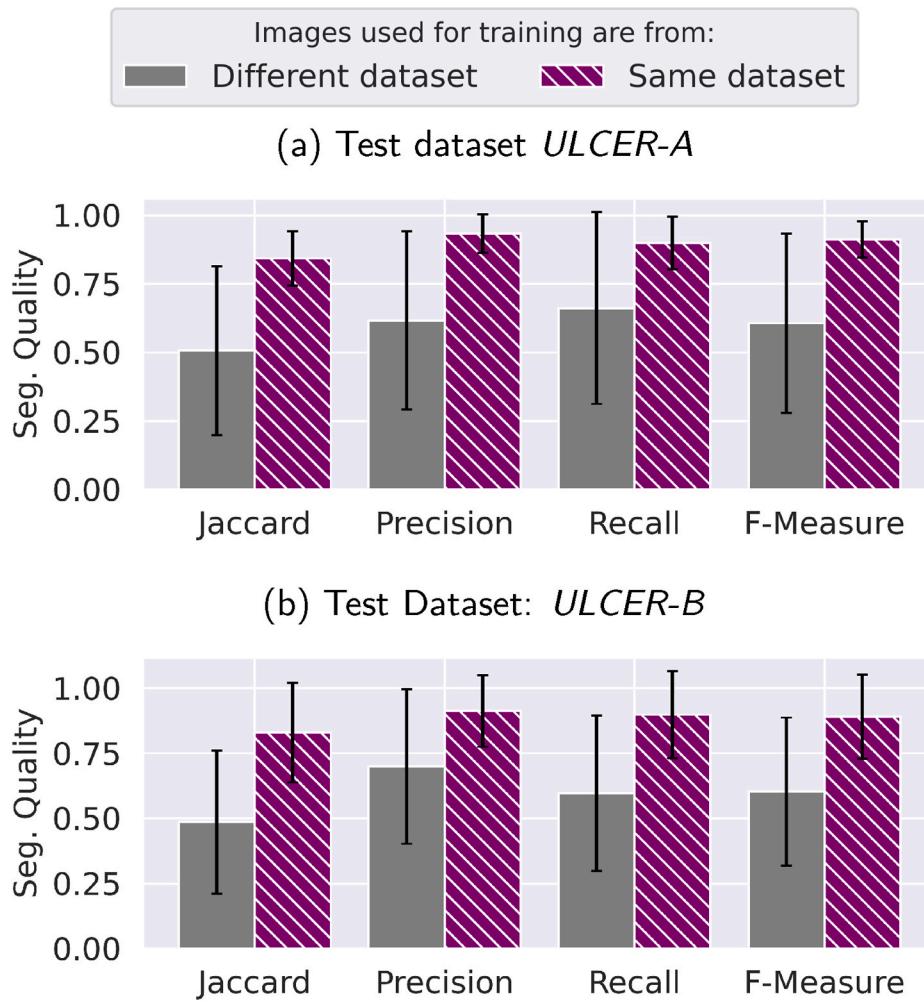


Fig. 13. The ASURA approach trained on different datasets: in (a), the models were used to segment images from *ULCER-A*. In (b), the models were used to segment images from *ULCER-B*. The built models do not generalize well for distinct image characteristics.

Table 5

Pixels per centimeter measurement. The automatic measurement performed by *UTrack* was better than the manual measurement in three out of five evaluated rulers.

| Image Type | GT | AM | Error (AM) | MM | Error (MM) |
|------------|----|----|------------|----|------------|
| Type i | 11 | 31 | 20 | 17 | 6 |
| Type ii | 12 | 20 | 8 | 13 | 5 |
| Type iii | 7 | 10 | 3 | 5 | 4 |
| Type iv | 30 | 29 | 1 | 34 | 14 |
| Type v | 24 | 23 | 1 | 25 | 9 |

GT: Ground truth.

AM: Automatic Measurement.

MM: Manual Measurement.

presented results comparable to ASURA, with 0.5% better F-Measure. *UTrack-Seg* also outperformed GrowCut by 22%, which is a region-growing segmentation approach.

In general, *UTrack-Seg* presented highly-accurate results, compared to ASURA, which is based on the UNet and has shown to be the competitor with the best results. Moreover, *UTrack-Seg* has the advantage of being unsupervised, so it does not require the expensive data augmentation and training steps of ASURA.

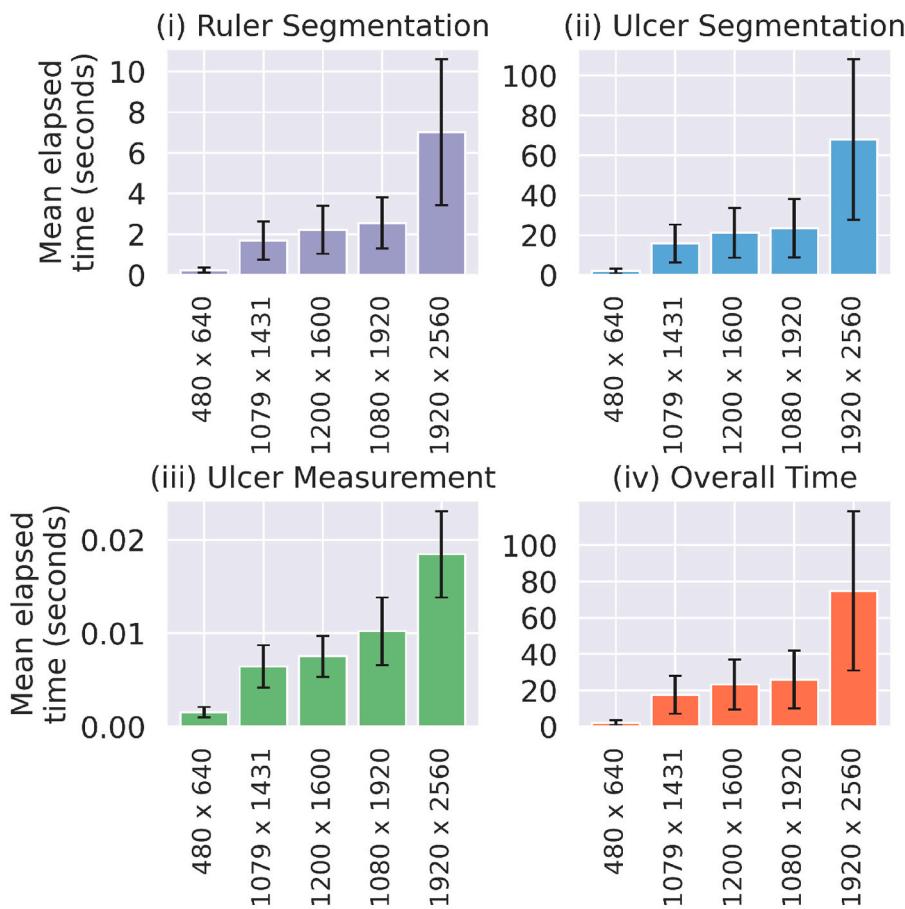
Fig. 11 shows four examples of images segmented by every evaluated approach. *UTrack-Seg* achieved the best quality results. Our method shows a more suitable adjustment to the wound edge, without gaps

(presented by most superpixel-based approaches), and with fewer over-segmented regions than the best competitor, ASURA. *UTrack-Seg* also had more favorable results than the region-growing competitor GrowCut, which did not properly adhere to the wound edge and segmented only portions of the wound.

In practice, health practitioners or patients often take pictures in environments with low or high luminosity. To evaluate *UTrack-Seg* in those scenarios, Fig. 12 shows the segmentation results according to different brightness levels. In part (a) we added or removed different proportions of brightness from every original image. In part (b) we ran *UTrack-Seg* and ASURA (the closest competitor) over every image and collected the segmentation results. *UTrack-Seg* has shown consistent segmentation results, mainly for high levels of brightness. The method receives as input the ulcer's exterior annotation, which guides the algorithm during the segmentation. The results indicate that *UTrack-Seg* was able to correctly detect the wound borders, even for very dark or bright images. *UTrack-Seg* takes the external outline of the wound (manually informed), and iterates inside the delimited region to segment the ulcer. Accordingly, when the image is very dark, the method may over segment the wound, but the number of pixels segmented will not be bigger than the outlined region, consequently limiting the segmentation error. However, notice that it would be a situation that a closer action of the user can easily fix.

ASURA was negatively impacted by the influence of low levels of brightness and generated low-quality segmentation results with high levels of brightness. This result was due to ASURA being trained only on

(a) Mean elapsed time for every segmentation task, considering 5 image resolutions



(b) Mean elapsed time from the three smartphones described in Table 3

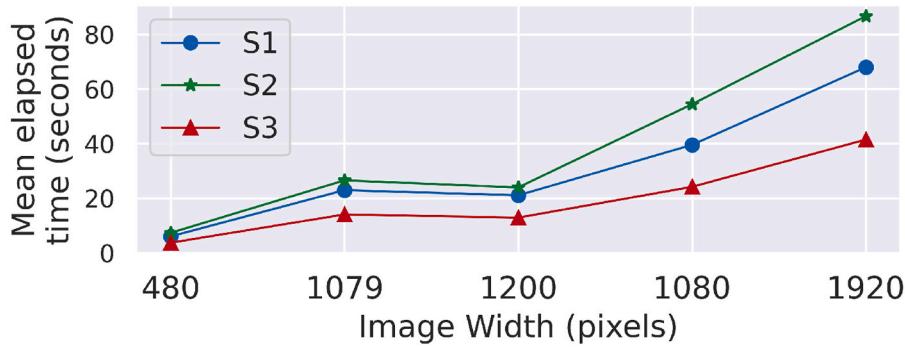


Fig. 14. Execution time of segmentation and measurement tasks using *UTrack-App*.

the original image dataset. Thus, the results indicate that the ASURA model needs to be trained on different brightness levels to improve its segmentation quality. In fact, ASURA and the other supervised competitors suffer from generalizing the segmentation model. To illustrate this characteristic, Fig. 13 shows ASURA's segmentation results when we evaluate its generalization error. In this example, we show ASURA's segmentation quality when trained and tested on the same dataset (purple dashed bars) or trained on a dataset and tested using the other one (gray bars). In (a), the gray bars correspond to images from *ULCER-A* segmented using a model trained on *ULCER-B*. The purple bars

correspond to images from *ULCER-A* segmented using a model trained on the same dataset. In (b), the gray bars correspond to images from *ULCER-B* segmented using a model trained on *ULCER-A*. The purple bars correspond to images from *ULCER-B* segmented using a model trained on the same dataset. For the purple bar experiments, we employed the same k-fold methodology described for the previous experiments. For the gray bar we used all images in *ULCER-A* and tested in every image in *ULCER-B*, and vice-versa. The results show that the segmentation quality heavily drops when the model is not trained on images that are similar to those being classified, lacking generalization. Accordingly, the use of

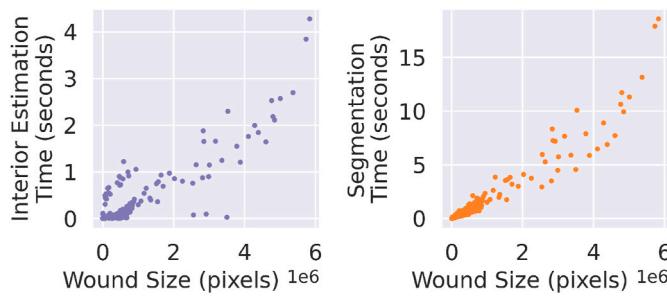


Fig. 15. Segmentation time according to the wound size, where each dot corresponding to an image: the closer to the left axis the quicker the segmentation.

ASURA in practice would require constantly feeding new images and expert-annotated masks to update the model and maintain the segmentation quality.

6.3. Tick detection

Table 5 presents the average number of pixels per centimeter measured by the Ground Truth (GT), Automatic Measurement (AM), and the Manual Measurement (MM), with their respective errors.

The manual measurements were performed by four subjects who were familiar with the task. The results were different for each of the five types of measurement tools employed for this validation (see Fig. 5). Values highlighted in gray indicate the best result obtained by each type of tool. AM resulted in high absolute errors for measurement tools of type i and ii. However, the method obtained consistent results for measurement tools of type iii, iv, and v. In contrast, MM reduced our method's error by up to 70% and 37.5% for types i and ii, where *UTrack* had difficulties establishing an accurate measure. For tools iii, iv, and v *UTrack* achieved more favorable results than MM did. Thus, practical scenarios indicate the need for a manual measurement option to improve wound measurement for some of the objects or rulers used for reference.

6.4. Performance analysis

In this subsection, we show *UTrack*'s performance. Fig. 14 (a) shows the execution time spent by *UTrack-App* to (i) segment the measurement tool, (ii) segment the ulcer, (iii) measure the wound size, and (iv) the overall time to process all tasks (i.e. i + ii + iii). We evaluate the elapsed times according to the image resolution. For each resolution setting, *UTrack* processed 10 images 15 times each and computed the average elapsed time. The segmentation of the measurement tool is faster than the ulcer segmentation because, in general, the ulcer region is bigger. The (iii) ulcer measurement includes the ticks' detection and wound area calculation and executes in less than a second. The complete execution (iv), even over the highest image resolution, takes a little more than a minute. As the charts show, *UTrack-App* can finish processing a 4K image (1920×2560 pixels), on average, in 75 s, and much less for images with smaller resolution. Considering the currently most used resolution in smartphones (i.e. 1080×1920 pixels), the app runs all tasks in less than 25 s, on average.

Fig. 14 (b) shows the average execution time to segment input images of different resolutions collected from three smartphones (see Table 3). In general, the higher the resolution, the higher the elapsed time spent by *UTrack*. Smartphone S2 finished the segmentation task in up to 87 s for 4K images, which was the longest execution time. The faster smartphone was S3, which took up to 41 s to segment 4K images and up to 3.7 s to segment images with the other resolutions. We also measured the execution time of ASURA to segment the images. As mentioned before, ASURA always takes the input image and resizes it to a 512×512 matrix before performing the segmentation. Accordingly,

the execution time is almost constant, regardless of the image resolution. ASURA segmented the images in up to 9.7 ± 1 s.

As *UTrack-Seg* iterates over the wound and ruler regions during the segmentation, we analyze the impact of the wound size on the execution time. Fig. 15 shows the interior estimation and the segmentation times according to the number of pixels corresponding to the wound size in the image. Even for high-resolution images with wound sizes bigger than 46k pixels, the interior estimation step of *UTrack-Seg* converges in less than 5 s. Considering the application scenario, where the app user expects timely analysis results, *UTrack-App* has shown to be well-fitted for the segmentation of dermatological wounds in a mobile application.

7. Conclusion

In this work, we proposed the *UTrack* framework to support semi-automatic wound measuring and tracking using mobile devices. The framework provides tools for patients, caregivers, and health professionals to measure ulcers, store historical data, visualize measurement results, and track dermatological wound healing evolution.

The health professional or caregiver can use *UTrack-App* to take a photo of the ulcer and inform the patient's symptoms related to the wound, such as pain and itch. The segmentation method *UTrack-Seg* takes the picture, delineates the outer outlines of the ulcer and of the reference object (e.g., a ruler) and segments both regions, automatically estimating the wound area in cm^2 or in^2 . The framework stores the new picture, symptoms, and the analysis results in a local database. If needed, when the automatic estimation happens to be imprecise, the user can manually input a measurement unit (cm or in). The algorithm uses this information to improve the estimation. The mobile application *UTrack-App* assists the users in visualizing the healing evolution, providing charts with the wound size evolution along time and the frequency of reported symptoms.

UTrack-Seg has the advantage of being unsupervised, requiring only a rough outline around the wound region. The studies show that *UTrack-Seg* segmented dermatological ulcers with F-Measure of 0.9, on average. The proposed segmentation method outperformed its competitors, presenting consistent and accurate results to assist health professionals in tracking dermatological ulcer evolution. Moreover, it is fast enough to be helpful in daily health care practice. For example, *UTrack-App* can execute the ruler segmentation, wound segmentation, and ulcer measurement tasks typically in less than 30 s when using the most common image resolution of 1080×1920 pixels.

UTrack is also dependable. *UTrack* outperformed the manual measurement of a centimeter in 3 out of 5 evaluated measurement tools. A manual measurement estimation can be used to correct the cases where the method is not precise enough. Importantly, *UTrack-App* does not require internet connection nor data transfer, performing all tasks locally (on the mobile device). However, when required, a user can generate a report and share it with a health professional using some communication applications. All images, related information, and results are stored on the mobile device, without patient identification, for privacy reasons. A subject of future work is to improve the framework in a version to allow professionals to manage and visualize data from several subjects and provide evolution prognostics and treatment directions. However, such features are outside the scope of this paper since such a tool requires proper approval by ethics committees and is intended only for health care professionals. We also intend to explore augmented reality (AR) technology to take advantage of mobile cameras' technical parameters as an alternative to measure the wound area.

Author contributions

All authors participated in the article preparation and review. Mirela T. Cazzolato, Jonathan S. Ramos, Lucas S. Rodrigues, Lucas S. Scabora and Daniel Y. T. Chino contributed to the design, implementation and validation of the proposed methods and were responsible for performing

the reported evaluations. Ana E. S. Jorge assisted in the design of the proposed methods, the validation assessment of the results in the medical domain, and the applicability of the current study. Paulo M. de Azevedo-Marques, Caetano Traina Jr. and Agma J. M. Traina assisted in the design and validation of the proposed methods. The authors declare that they have read and approved the final version of this article.

Declaration of competing interest

The authors declare that there is no competing interest regarding the work reported in this paper.

Ethical approval

The images used in this study were obtained with the permission of a researcher involved in a clinical trial approved by the ethical research committee of Ribeirao Preto Medical School, from the University of Sao Paulo.

Acknowledgments

This research was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, by the São Paulo Research Foundation (FAPESP, grants No. 2016/17078-0, 2020/07200-9, 2018/24414-2, 2020/11258-2, 2017/23780-2, 2020/10902-5, 2016/17330-1), and the National Council for Scientific and Technological Development (CNPq).

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.combiomed.2021.104489>.

References

- [1]
- [2]
- [3]
- [4]
- [5]
- [6]
- [7]
- [8]
- [9]
- [10]
- [11]
- [12]
- [13]
- [14]
- [15]
- [16]
- [17]
- [18]
- [19]
- [20]
- [21]
- [22]
- [23]
- [24]
- [25]
- [26]
- [27]
- [28]
- [29]
- [30]
- [31]
- [32]
- [33]
- [34]

[35

[36

[37

[38

[39